

RASTREAMENTO O UTILIZANDO COMUNICAÇÃO



**DANIELLY KARYNE SCHRAMMEL, FÉLIX HOFFMANN SEBASTIANY,
GIAN PABLO RISKE, MATEUS DIEL KUHN**

O que é LoRa??



LoRa

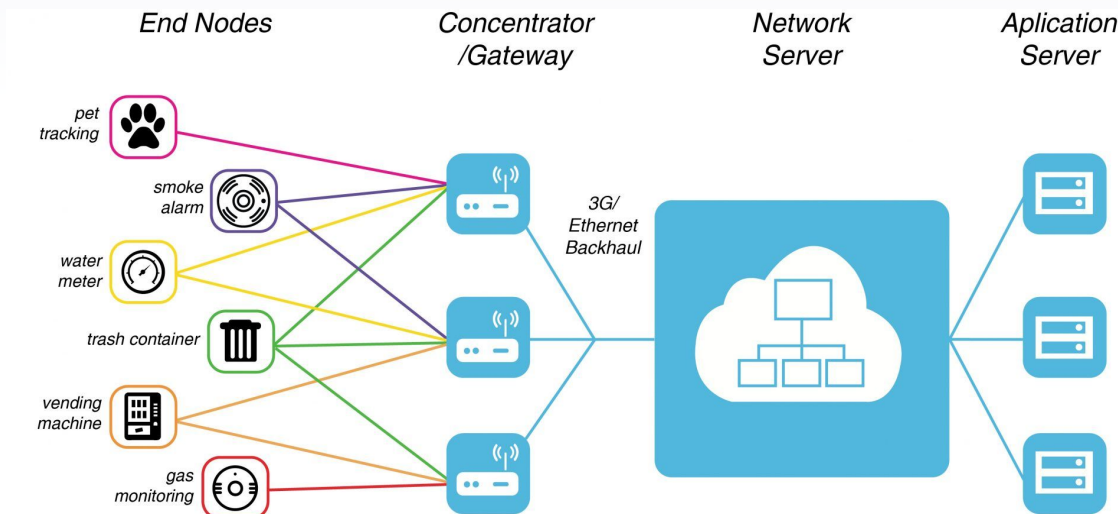
Long Range

Longo Alcance

A tecnologia LoRa é uma nova forma de comunicação sem fio, semelhante ao Wifi e ao Bluetooth mas que permite comunicações em longas distâncias (tipicamente 3 a 4 quilômetros em área urbana), gastando muito pouco de energia.

Rede LoRa

- ▶ Topologia em Estrela
- ▶ Elementos Básicos:
 - End-Nodes
 - Gateways
 - Network Server
 - Application Server



Aplicações que usam LoRa

Sensores de Temperatura

Uma empresa de Brasília desenvolveu um sensor de temperatura integrando um módulo LoRa e monitora câmaras frias, geladeiras, e sistemas de refrigeração diversos e os envia para a nuvem de forma simples e eficiente usando a tecnologia LoRa.

Como uma câmara fria é extremamente lacrada, tecnologias como Wifi e GSM não permitem enviar os dados para a nuvem, ao contrário do LoRa que funciona muito bem para atravessar paredes.

Medidores de Energia

Com o LoRa é possível capturar dados de consumo de clientes residenciais ou mesmo rurais enviá-los para o data center das companhias de energia de forma bem simples sem necessidade de interação humana.

Telemetria para Elevadores

Quando há picos de energia vários elevadores podem parar de funcionar e para reiniciá-los é necessário a ida de técnicos até os locais para apertar apenas uma tecla.

Como o LoRa funciona muito bem em ambientes ruidosos como é o caso de um elevador, é possível extrair todos os dados do mesmo, enviar comandos de forma remota e assim reiniciar o equipamento sem a necessidade da presença do técnico. Isso acarreta economia de tempo e dinheiro para as empresas de manutenção de elevadores.

Aplicações que usam LoRa

Rastreamento

Com o LoRa é possível receber dados de localização de um GPS, e enviar as coordenadas para o Gateway, que envia para o Network Server, que envia para aplicação. Caracterizando um sistema de rastreamento com baixo consumo de energia.

Pseudocódigo

● Gateway

```
void TaskIcode( void * pvParameters ) {
    for (;;) {
        if (fifo_n_data >= maxToSent) {
            while (fifo_n_data != 0) {
                String ms = fifo_pull();
                post += String(msg) + "=" + ms + "&";
                bpFifo[msg] = ms;
                msg++;
            }
            if (enviaServidor(post.substring(0, post.lastIndexOf("&")))) {
                Serial.println("Enviado com sucesso para o banco de dados!");
            } else {
                for (int i = 0; i < FIFO_MAX; i++) {
                    if (bpFifo[i].length() != 0) {
                        fifo_push(bpFifo[i]);
                        bpFifo[i] = "";
                    } else {
                        break;
                    }
                }
            }
        }
    }
}
```

```
void onReceive(int packetSize) {
    if (packetSize == 0) return;
    int recipient = LoRa.read();
    byte sender = LoRa.read();
    byte incomingLength = LoRa.read();
    String incoming = "";
    while (LoRa.available()) {
        incoming += (char)LoRa.read();
    }
    if (incomingLength != incoming.length()) {
        return;
    }
    if (recipient != localAddress && recipient != 0xFF) {
        return;
    }
    if (fifo_data_isfull() == 1) {
        sendOk("Congestionamento!", sender);
    } else {
        fifo_push("0x" + String(sender, HEX) + ";0x" + String(localAddress, HEX) + ";" + incoming);
        sendOk("OK, Gateway 0xBB recebeu a mensagem!", sender);
    }
    Serial.println();
    piscaLed(porta);
}
```

Pseudocódigo

- Node's

```
void Task1code( void * pvParameters ) {
    for (;;) {
        if (millis() - lastSendTime > interval) {
            if (gps.location.isValid() && gps.location.isUpdated() && TinyGPSPlus::distanceBetween(gps.location.lat(), gps.location.lng(), 0.00000, 0.00000) > 0.00) {
                piscaLed(ledGPS);
                String mensagem = String(gps.location.lat(), 6) + ";" + String(gps.location.lng(), 6) + ";" + printDateTime(gps.date, gps.time);
            }
            lastSendTime = millis();
        }
        onReceive(LoRa.parsePacket());
    }
}

void Task2code( void * pvParameters ) {
    for (;;) {
        while (Serial2.available()) {
            gps.encode(Serial2.read());
        }
        delay(1);
    }
}
```

Pseudocódigo

- Node's

```
void sendMessage(String outgoing) {
    LoRa.beginPacket();
    LoRa.write(destination);
    LoRa.write(localAddress);
    LoRa.write(outgoing.length());
    LoRa.print(outgoing);
    LoRa.endPacket();
}

void onReceive(int packetSize) {
    if (packetSize == 0) return;    int recipient = LoRa.read();    byte sender = LoRa.read();
    byte incomingLength = LoRa.read(); | String incoming = "";
    while (LoRa.available()) {
        incoming += (char)LoRa.read();
    }
    if (incomingLength != incoming.length()) {
        return;
    }
    if (recipient != localAddress && recipient != 0xCC) {
        return;
    }
    if (incoming.indexOf("OK") > -1) {
        piscLed(ledLora);
    }
}
```


Banco de Dados

node:

- id (BIGINT(20))
- id_node (VARCHAR(10))
- id_server (VARCHAR(10))
- latitude (VARCHAR(13))
- longitude (VARCHAR(13))
- date_node (VARCHAR(13))
- time_node (VARCHAR(10))
- date_server (VARCHAR(13))
- time_server (VARCHAR(10))

usuario:

- id (INT(10))
- nome (VARCHAR(70))
- senha (VARCHAR(40))
- login (VARCHAR(50))

dcalendario:

- id (INT(11))
- Data (DATE)
- Semana_Ano (INT(11))
- Semana_Mes (INT(11))
- Dia_Semana (TEXT)
- Mes_Nro (INT(11))
- Mes (TEXT)
- Mes_Abrevido (TEXT)
- Trimestre (INT(11))
- Trimestre_Abrevido (TEXT)
- Ano (INT(11))

► Materiais Utilizados

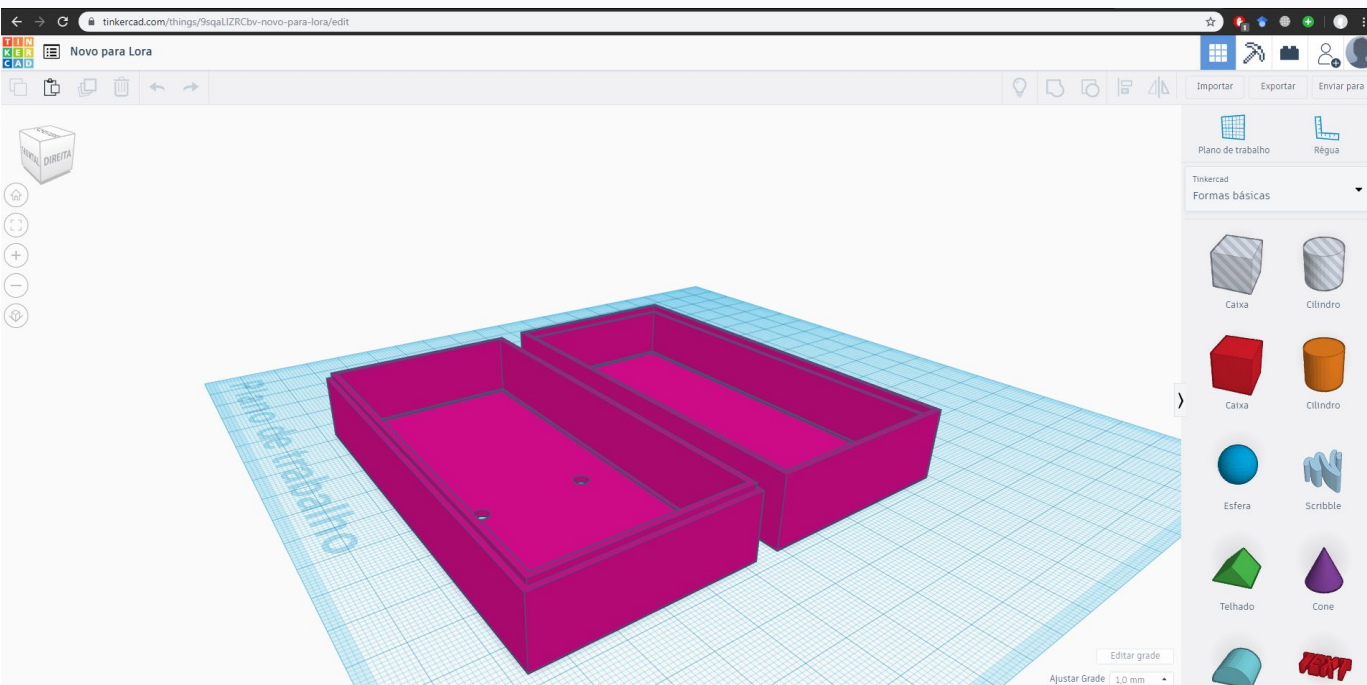
Hardware

- ▶ Gateway: ESP32, LoRa, Led, bateria 18650, Switch on/off, Case
- ▶ End-Nodes: ESP32, LoRa, Módulo GPS, Led, bateria 18650, Switch on/off, Case

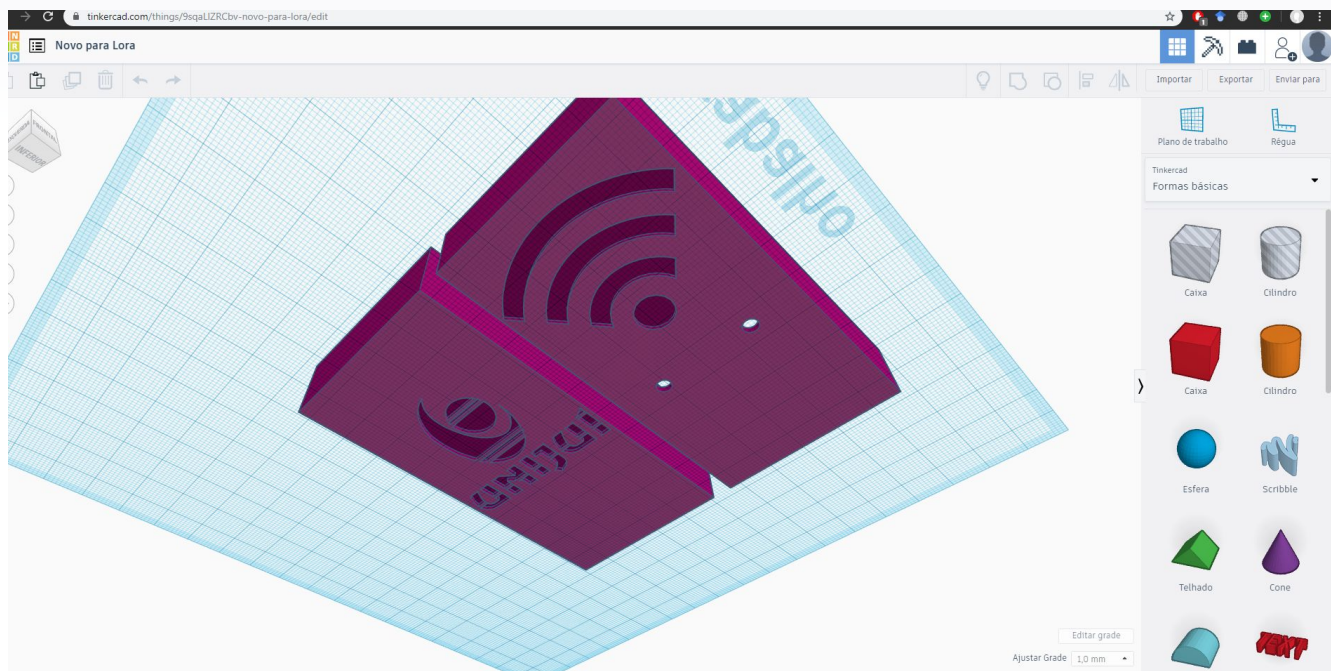
Software

- ▶ MySQL Workbench
- ▶ Arduino IDE
- ▶ Microsoft Power BI
- ▶ Autodesk TinkerCad

Projeto do Case para o protótipo



Projeto do Case para o protótipo



Produto Final



Dashboard Power BI

